

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

Wesley Nascimento Araujo

Estudo comparativo de plataformas para desenvolvimento de soluções IoT

Juiz de Fora

2021

Wesley Nascimento Araujo

Estudo comparativo de plataformas para desenvolvimento de soluções IoT

Trabalho de conclusão de curso apresentado ao Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação.

Aprovada em 02 de Setembro de 2021

BANCA EXAMINADORA

Prof. Dr. Eduardo Pagani Julio - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Marcelo Ferreira Moreno
Universidade Federal de Juiz de Fora

Prof. Dr. Eduardo Barrere
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Agradeço aos meus pais e familiares por nunca me deixarem desistir.

Agradeço ao professor Eduardo Pagani pela orientação, amizade e paciência durante o desenvolvimento do trabalho e no decorrer do curso.

Agradeço também a todos os professores que nos proporcionaram algum conhecimento em suas disciplinas.

Agradeço aos meus amigos que sempre me acompanharam durante a jornada, deixando ela um pouco mais suave.

RESUMO

No mundo atual, Internet das Coisas (*Internet of Things - IoT*) é um conceito de grande influência que vem crescendo cada vez mais no ramo da tecnologia e por isso a maioria das empresas precisam buscar desenvolver seus sistemas pensando em soluções IoT que permitam acesso e gerenciamento a um conjunto de informações relevantes, independente do lugar que elas estejam. Atualmente, existem diversas ferramentas distribuídas como PaaS (*Plataform as a Service*) que facilitam o trabalho de criar uma solução IoT capaz de tudo, desde medir a temperatura de uma casa, rastrear um veículo, até controlar a iluminação da residência através de um *desktop* ou um *smartphone*. Existem vários desafios ao escolher a melhor plataforma para utilizar em cada sistema. O trabalho descrito a seguir realizou um estudo comparativo em cima dessas plataformas disponíveis para soluções IoT e através dele apresentou vantagens e desvantagens entre elas, indicando possíveis soluções que podem ser desenvolvidas.

Palavras-chave: IoT, PaaS, plataformas IoT.

ABSTRACT

In the current world, Internet of Things (*Internet of Things - IoT*) is a concept of great influence that has been growing more and more in the field of technology and therefore most companies need to seek to develop their systems thinking about IoT solutions that allow access and management to a set of relevant information, regardless of where they are. Currently, there are several distributed tools such as PaaS (*Platforms as a Service*) that facilitate the job of creating an IoT solution capable of everything from measuring the temperature of a home, tracking a vehicle, to controlling the lighting of the home by through a *desktop* or a *smartphone*. There are several challenges when choosing the best platform to use on each system. The work described below carried out a comparative study on these platforms available for IoT solutions and through it presented advantages and disadvantages among them, indicating possible solutions that can be developed.

Keywords: IoT, PaaS, IoT plataforms.

LISTA DE ILUSTRAÇÕES

Número de plataformas conhecidas (Lueth et al, 2019)	10
Arquitetura IoT-Framework	13
<i>End Devices</i>	14
Protocolo MQTT (Torres et al, 2016)	16
Tago.IO	24
Criação de dispositivos na Tago.IO	25
Dispositivo na Tago.IO	26
Conectando ao <i>broker</i>	26
Inscrivendo no tópico	27
Enviando informação para o tópico	27
<i>Action</i> para envio e-mail	28
<i>Dashboard</i> Tago.IO	28
<i>Services</i> Tago.IO	29
ThingSpeak	30
<i>New Channel</i> ThingSpeak	31
<i>Create Device</i> ThingSpeak	31
<i>Connection</i> ThingSpeak	32
<i>Subscribe Topic</i> ThingSpeak	32
<i>Topic & payload</i> ThingSpeak	32
<i>Dashboard</i> ThingSpeak	33
<i>React</i> ThingSpeak	34
<i>Services</i> ThingSpeak	34
Azure IoT Hub	35
Criação <i>Resource</i> Azure	36
Criação dispositivo Azure	36
Configurar novo dispositivo Azure	36
<i>Connection String</i> Azure	37
Geração <i>token</i> Azure	37
<i>Connection broker</i> Azure	38
Inscrição tópico Azure	38
Enviando informação para o tópico	39
<i>Endpoint Key Azure Function</i>	39
<i>Local Settings Function</i>	39
<i>Azure Function</i>	40
<i>Dashboard</i> Azure	40
<i>Services</i> Azure	41

LISTA DE TABELAS

Tabela 1 – Comparação entre as plataformas	23
--	----

LISTA DE ABREVIATURAS E SIGLAS

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
IoT	<i>Internet of Things</i>
PaaS	<i>Platform as a Service</i>
IaaS	<i>Infrastructure as a Service</i>
M2M	<i>Machine to Machine</i>
UI	<i>User Interface</i>
MQTT	<i>Message Queue Telemetry Transport</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
RSSF	Redes de Sensores Sem Fio
SDK	<i>Software Development Kit</i>
API	<i>Application Programming Interface</i>
SMS	<i>Short Message Service</i>
IDE	<i>Integrated Development Environment</i>
OS	<i>Operating System</i>
AWS	<i>Amazon Web Services</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
LoRaWAN	<i>Long Range Wide Area Network</i>

SUMÁRIO

1	INTRODUÇÃO	9
1.1	JUSTIFICATIVA E CARACTERIZAÇÃO DO PROBLEMA	9
1.2	OBJETIVOS	11
2	REVISÃO BIBLIOGRÁFICA E FUNDAMENTAÇÃO TEÓRICA	12
2.1	PROBLEMAS AO ESCOLHER UMA PLATAFORMA	12
2.2	IMPORTÂNCIA NO PADRÃO DE UMA ARQUITETURA	12
2.3	ARQUITETURA DE UMA PLATAFORMA IOT	13
2.4	PROCESSAMENTO DE DADOS COM FOCO EM PLATAFORMAS IOT E EM BORDA	15
2.5	PROTOCOLO MQTT	15
3	ESTUDO DE PLATAFORMAS IOT DO MERCADO	17
3.1	TAGO.IO	17
3.2	THINGSPEAK	17
3.3	MICROSOFT AZURE IOT HUB	18
3.4	GOOGLE CLOUD PLATAFORM	19
3.5	IBM BLUEMIX	19
3.6	THINGWORX	19
3.7	DIGITAL SERVICE CLOUD	20
3.8	ZETTA	20
3.9	NIMBITS	21
3.10	AMAZON WEB SERVICES	21
4	ESTUDO DE CASO	24
4.1	IMPLEMENTAÇÃO BÁSICA EM PLATAFORMAS IOT	24
4.2	PLATAFORMA TAGO.IO	24
4.3	PLATAFORMA THINGSPEAK	30
4.4	PLATAFORMA AZURE IOT HUB	35
5	CONCLUSÃO E TRABALHOS FUTUROS	42
	REFERÊNCIAS	44
	Referências Bibliográficas	44

1 INTRODUÇÃO

Com a evolução da tecnologia, o poder de processamento aumentou em todos os tipos de *hardwares*, a necessidade de comunicação via *Internet* por meio de coisas, sejam elas portáteis ou não, foi inevitável. Atualmente, a quantidade de dispositivos existentes no Brasil superam a quantidade da população, como visto na pesquisa FGVcia (2021), ou seja, grande parte dos humanos possuem *smarthphones*, geladeiras, carros e até mesmo casas que transmitem dados para rede.

Para entender Internet das Coisas (*Internet of Things* - IoT), primeiramente é importante compreender o conceito de Internet dos Humanos. Basicamente, para haver conexão com algum serviço de interesse é necessário um dispositivo que seja capaz de se comunicar com um provedor de conectividade, que pode ser via cabo ou rede móvel e assim retornar informações para o interessado. No caso de Internet das Coisas, se trata de objetos conectados por meio de alguma rede capaz de prover *Internet* para eles, desta forma o mesmo se torna capaz de se conectar a um determinado serviço e assim executar as devidas funções que o foram designadas.

Neste ponto é onde surge a necessidade de um trabalho que forneça uma conectividade padronizada, segurança e integração com aplicativos para o dispositivo. Para isso existem as Plataformas de Serviço IoT (*Plataform as a Service* - PaaS) que podem ser definidas como um ambiente onde se pode interagir de forma simples, efetiva e descomplicada com tudo, desde rastrear um veículo, monitorar o volume de uma televisão, saber se um portão está aberto até automatizar o desligamento de aparelhos elétricos.

Uma plataforma IoT está conectada a *Internet* e é composta basicamente por uma interface *web*, para a comunicação com o usuário, banco de dados para armazenamento e uma interface para se comunicar com dispositivos ou sensores que fornecem medições para que essas plataformas apresentem um conjunto de informações para o usuário. Esses dispositivos podem ser desenvolvidos a partir de microcontroladores, placas de prototipagem como arduino, Raspberry Pi, entre outros.

1.1 JUSTIFICATIVA E CARACTERIZAÇÃO DO PROBLEMA

Hoje em dia, a quantidade de aparelhos com conexão a *Internet* tem crescido rapidamente. Além disso, se tratando de tecnologia, como apresentado no relatório da Ericsson HEUVELDOP (2017) a evolução se apresenta de forma constante, e ainda temos o fato de que os seres humanos sempre buscam maneiras de facilitar e modernizar seu estilo de vida. Por conta disso, ao realizar a construção de um sistema, deve-se sempre pensar no fato de que a tecnologia nunca para de evoluir e que seus usuários, na maioria das vezes, buscam sistemas que os proporcionem facilidade e automação tecnológica no seu dia a dia. Dessa forma o desenvolvimento de um sistema pensando em um serviço IoT

se torna cada vez mais necessário.

Quando só haviam as soluções de Máquina para Máquina (*Machine to Machine - M2M*), os dispositivos eram conectados entre si ou a *softwares* próprios com protocolos específicos projetados para um determinado fim, utilizavam paradigmas de aplicações tradicionais, onde eram instalados e executados em computadores ou servidores locais, o que os deixavam limitados quanto a configurações de *hardware*, Sistemas Operacionais (*Operating System - OS*) e *softwares*.

Com o advento da computação em nuvem (*cloud computing*) e o modelo PaaS os *softwares* dedicados passaram a ser utilizados como serviço *online* por diversos clientes em uma única infraestrutura. O avanço desse modelo beneficiou a Internet das Coisas, pois ele provê toda a infraestrutura desde o *hardware*, sistema operacional e conectividade até um nível de aplicação específico de maneira escalável e com segurança.

Atualmente, como levantado em um estudo realizado pelo IoT Analytics, (Lueth et al, 2019) o crescimento de plataformas IoT de 2015 até 2020, ilustrado na Figura 1, teve um valor considerável e possui tendência a aumentar. O artigo também apresenta que atualmente existem cerca de 620 plataformas conhecidas publicamente no mercado. Então como saber qual dessas plataformas escolher? Para isso pode-se levantar alguns estudos sobre elas, analisar o que cada plataforma proporciona e quais são suas vantagens e desvantagens.

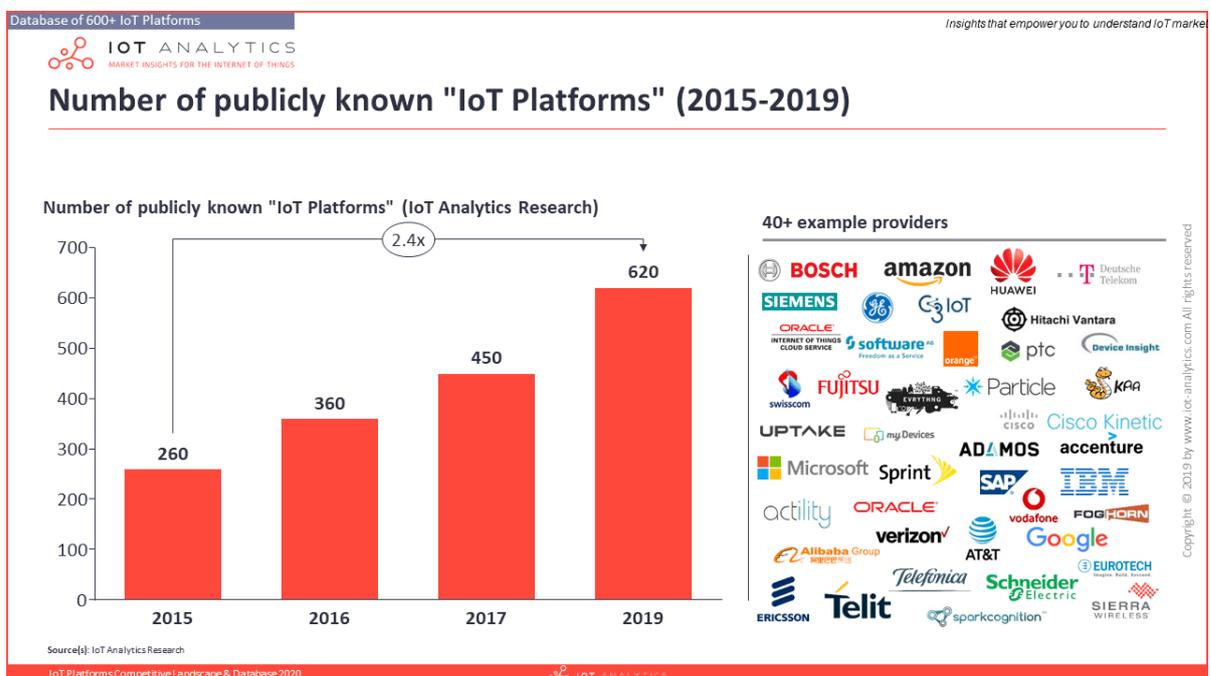


Figura 1: Número de plataformas conhecidas (Lueth et al, 2019)

1.2 OBJETIVOS

Esse trabalho visa comparar diferentes plataformas que são utilizadas para desenvolvimento de soluções IoT, como por exemplo, Tago.IO, ThingSpeak, Microsoft Azure IoT Hub, Amazon Web Services IoT Platform, entre outras. Criando um estudo de caso e a partir do mesmo, apresentar o que cada solução dispõe, identificando e comparando seus pontos positivos e negativos, para auxiliar um possível usuário em sua tomada de decisão sobre qual delas escolher para o desenvolvimento de uma melhor solução IoT. Além disso, o trabalho também deseja mostrar para os leitores que uma plataforma é diferente de outra e a escolha de uso, não deve ser pela mais famosa e sim pela que apresenta a melhor solução para um problema específico.

2 REVISÃO BIBLIOGRÁFICA E FUNDAMENTAÇÃO TEÓRICA

2.1 PROBLEMAS AO ESCOLHER UMA PLATAFORMA

Quando se trata de IoT temos que o objetivo geral é interconectar o mundo físico com o digital. Sendo assim, o mundo físico precisa ser medido e traduzido em dados para o computador, bem como, o mundo digital deve ser medido e traduzido para os seres humanos. No artigo (Guth et al, 2016), os autores mencionam o aumento de conhecimento voltado para área de IoT e, conseqüentemente, o aumento na quantidade de plataformas IoT. Esse aumento acaba levando à problemas de compreensão, que podem ocorrer durante o projeto até a seleção de uma solução adequada. O objetivo desse artigo é mostrar que além de questão de desempenho a variedade de plataformas pode acabar gerando um problema de compreensão por parte de quem está desenvolvendo e por isso os autores apresentam uma arquitetura de referência com base em diferentes plataformas IoT que fornece uma base uniforme para entender, comparar e avaliar diferentes soluções de IoT.

No artigo (Salami et al, 2018) os autores descrevem a importância de fazer o balanço entre vantagens e desvantagens das diferentes distribuições de PaaS para IoT e assim escolher a que mais se adapta a solução buscada. Os autores citam alguns pontos que devem ser observados ao fazer a escolha da plataforma, tais como, gerenciamento e monitoramento de dados, baixa perda de informações, velocidade, baixa latência, entre outros. O objetivo do texto é fazer testes de diferentes plataformas e investigar os critérios qualitativos e quantitativos das mesmas, para isso os autores mantêm o foco em algumas mais específicas, como, por exemplo, Thingspeak, Xively e AWS IoT e realiza os testes em diferentes cenários, porém, com as mesmas condições.

2.2 IMPORTÂNCIA NO PADRÃO DE UMA ARQUITETURA

O texto descrito pelos autores (Pires et al, 2015) descreve o fato de que ambientes IoT são caracterizados por um alto grau de heterogeneidade de dispositivos e protocolos de rede. Para tratar esse número de diferentes dispositivos várias plataformas de *middleware* para IoT tem sido propostas visando abstrair as diferenças desses dispositivos e promover a interoperabilidade entre eles. Entretanto, a falta de padronização existente em IoT faz com que tais plataformas adotem diferentes modelos de programação e não abordem adequadamente uma série de requisitos importantes. Dessa forma, outra ferramenta como as arquiteturas de referência podem ser usadas, pois, elas conseguem definir um conjunto inicial de blocos de construção para ambientes IoT. O objetivo do texto é, levantar e discutir requisitos de *middleware* para IoT, apresentar algumas arquiteturas de referência existentes e mostrar o uso prático de uma plataforma de IoT.

Entende-se que uma plataforma é capaz de distribuir dados que foram armazenados

por sensores para outras aplicações ou usuários através de uma Interface de Usuário (*User Interface* - UI). Ela funciona basicamente com três camadas, sendo elas, os sensores, a plataforma escolhida e os usuários/aplicações, um exemplo disso foi demonstrado no artigo (Vandikas et al, 2014), onde ele usou o IoT-Framework para um estudo de performance. O modelo do IoT-Framework pode ser visto na imagem a seguir:

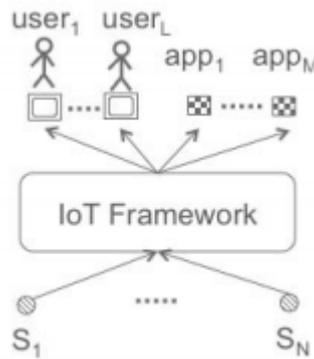


Figura 2: Arquitetura IoT-Framework

Os sensores operam de diversas maneiras diferentes em termos de padrões de comunicação, eles podem ter as tarefas de relatar certos eventos, relatar quando estiverem sendo pesquisados ou responder por comandos. O *framework* recebe os fluxos de dados dos sensores e os envia para a interface ou para as outras aplicações interessadas. A UI possibilita o usuário executar certos eventos de maneira rápida e prática. A ideia dos autores foi testar a estabilidade da plataforma, e eles usaram uma série de *hardwares* diferentes para isso.

É importante destacar que o trabalho de um desenvolvedor ao criar uma solução IoT, é um pouco complicado devido à indisponibilidade dos dispositivos físicos necessários para realizar testes de implementação. Devido a isso muitos desenvolvedores utilizam da ideia de usar emuladores para reproduzir esses *hardwares* de maneira virtual. Como observado no artigo (Sousa et al, 2019), a autora apresenta uma solução de um emulador de Publisher MQTT (*Message Queuing Telemetry Transport*), propondo a emulação de uma RSSF (Redes de Sensores Sem Fio), onde esse emulador possa gerar dados e publicá-los para um *broker* MQTT, disponibilizando os mesmos para aplicações interessadas.

2.3 ARQUITETURA DE UMA PLATAFORMA IOT

De um modo geral, uma solução IoT é composta por:

- Dispositivo / *end device*: sistema embarcado (*hardware* e *software*) que fica em campo, sendo responsável por coletar e, em alguns casos, pré-processar dados. É também responsável por enviar e receber os dados obtidos para uma plataforma

IoT ou um servidor remoto. O envio e recepção dos dados de uma plataforma IoT são feitos via *Internet* ou via rádio em conjunto com *gateways* como, por exemplo, *Bluetooth*, LoRaWAN (*Long Range Wide Area Network*), entre outros. Os sensores que podem ser de temperatura, umidade, pressão atmosférica e atuadores (LEDs, motores, *displays*) ficam junto a esses dispositivos.



Figura 3: *End Devices*

- Canal de comunicação: os *end-devices* enviam e recebem dados para uma plataforma por meio de um canal de comunicação, estes que podem ser:
 - *Internet* (cabeadada ou Wi-Fi): o dispositivo tem a capacidade e possibilidade de se conectar diretamente à *Internet* e se comunicar com a plataforma IoT.
 - Via *gateway*: o dispositivo não tem capacidade de se conectar diretamente com à *Internet*, logo precisa utilizar um *gateway* para se comunicar com uma plataforma. O *gateway* é um componente (*hardware* e *software*) que contém um sinal de rádio como, por exemplo, *Bluetooth* e LoRaWAN, pois os mesmos podem ser usados como uma camada de rede e garantir conectividade com à *Internet*, desse modo ele é responsável por passar os dados entre a plataforma e o *end-device*.
- Plataforma IoT: os dados coletados pelos dispositivos trafegam pelo canal de comunicação para chegar à *Internet*, onde estão destinados às plataformas IoT. As plataformas utilizam tais dados para gerar informações úteis. Ou seja, ela é o elemento responsável por gerar valor e permitir que o usuário tome as melhores decisões possíveis em uma solução IoT. Algumas tarefas comuns de plataformas são: armazenamento e processamento de dados, para os transformar em informações úteis, organização e exibição das informações em um formato humanamente intuitivo, tomar ações automáticas dependendo dos dados colhidos, garantir que as informações não sejam acessíveis por agentes externos, entre outros.

2.4 PROCESSAMENTO DE DADOS COM FOCO EM PLATAFORMAS IOT E EM BORDA

Quando se trata de uma solução IoT os dispositivos são os elementos que ficam em campo e são responsáveis pela coleta e geração dos dados que serão utilizados no contexto de Internet das Coisas. Dependendo da finalidade e poder computacional dos dispositivos, estes podem pré-processar muitas informações antes de realizar o envio para a plataforma, ou seja, enviar menos dados, porém com mais informações úteis. Esse tipo de uso dos dispositivos como partes integrantes do processamento dos dados colhidos é chamado de processamento com foco em borda (*Edge Analytics*). Esse tipo de processamento traz benefícios como, por exemplo, menor custo, menor tráfego de dados e menor tempo de tomada de decisão.

O processamento com foco nas plataformas IoT, ao contrário do *Edge Analytics*, não faz o pré-processamento dos dados utilizando o *end-device*. Eles são apenas coletores de dados e todo o processamento fica por conta da plataforma escolhida. Essa estratégia é mais usada quando as informações a serem geradas dependem de anos ou décadas de coleta, quando a estratégia do negócio é utilizar apenas de serviços em nuvem e quando os gastos são investidos nas plataformas em vez dos dispositivos. Porém, para este trabalho, queremos analisar o desempenho das plataformas, então será utilizado o modelo com foco de processamento nas plataformas.

2.5 PROTOCOLO MQTT

O MQTT (*Message Queue Telemetry Transport*) consiste em um protocolo de mensagens leve, criado para comunicação M2M. Com MQTT, é possível comunicar dispositivos diversos, indo desde dispositivos remotos até computadores eletrônicos com altíssimo poder computacional. A comunicação é feita com mensagens trafegadas em tópicos (chaves/identificadores), os quais informam para quem cada mensagem deve ser enviada. Por exigir muito pouco poder computacional e conectividade com *Internet*, este é um dos protocolos ideais para dispositivos embarcados com limitações de processamento, memória e banda de *Internet*. Por esta razão, o protocolo é muito usado e difundido em plataformas e projetos de Internet das Coisas.

Uma comunicação MQTT é composta basicamente por três partes:

- **Publishers:** dispositivos que publicam e enviam informações para o *broker*.
- **Subscribers:** dispositivos que se inscrevem e recebem informações do *broker*.
- **Broker:** responsável por receber as informações dos *Publishers* e direcioná-las aos devidos *Subscribers*, além de gerenciar a conexão de todos os clientes MQTT (dispositivos *publishers* e *subscribers*). Em resumo, o *broker* aguarda as mensagens

dos dispositivos e as redireciona a cada cliente MQTT que assinou o tópicos da referida mensagem. Portanto, essa parte tem o trabalho pesado de gerenciar toda a comunicação.

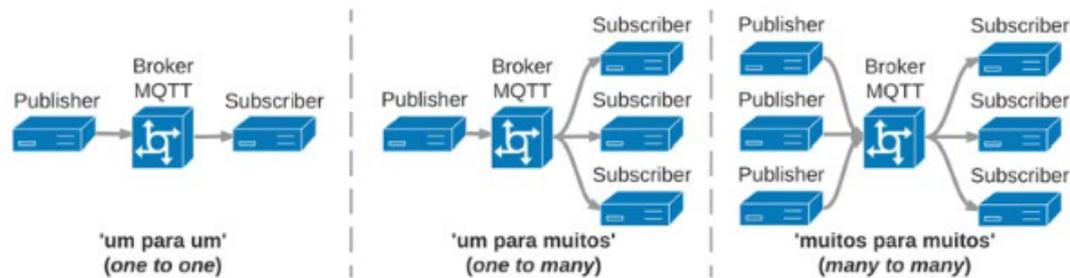


Figura 4: Protocolo MQTT (Torres et al, 2016)

As mensagens do protocolo possuem duas partes importantes:

- **Tópico:** informa quem deve receber tal mensagem.
- **Payload:** os dados a serem trafegados na mensagem.

Quando se trata de IoT esse é o protocolo ideal a ser utilizado, uma vez que ele é um protocolo leve, pode ser aplicado a quase todos os tipos de dispositivos, é versátil, então os dispositivos precisam conhecer somente seu endereço, não sendo preciso nenhuma abertura de portas ou acessos especiais no roteador, além disso ele possui SDKs (*Software Development Kit*) para várias linguagens de programação.

3 ESTUDO DE PLATAFORMAS IOT DO MERCADO

A escolha de uma plataforma IoT pode se tornar uma tarefa muito complicada devido a quantidade de ferramentas disponíveis. Considerando a complexidade dessa tomada de decisão, foi feita uma pesquisa pelo artigo (Nakhuva et al, 2015) que visa estudar algumas das plataformas existentes no mercado.

O texto começa dando uma introdução do tema IoT e aponta algumas diferenças entre “*Platform IoT*” e “*Middleware IoT*”. Ele descreve *Platform IoT* como um conjunto completo de serviços que facilitam atividades como desenvolvimento, implantação, manutenção e análises, bem como recursos de tomada de decisão inteligente para uma aplicação IoT. Por outro lado, *Middleware IoT* é um conjunto de serviços que se destina principalmente a superar o problema de heterogeneidade de todo o sistema IoT, permitindo uma comunicação suave entre dispositivos e componentes de diferentes fornecedores e diferentes tecnologias.

Após essa breve introdução no aspecto geral do tema, o artigo apresenta a variedade de plataformas disponíveis e destaca algumas das mais famosas.

3.1 TAGO.IO

A Tago.IO oferece uma plataforma de nuvem de ponta que transforma a maneira como as empresas criam valor a partir de produtos conectados e interações do usuário. Excelente para desenvolver protótipos uma vez que possui planos gratuitos que permitem envio de *e-mails*, SMSs, armazenamento exclusivo para cada dispositivo criado (*Device Buckets*) e análises executadas por *scripts* do usuário. Algumas características dessa plataforma:

- Escalabilidade;
- Armazenamento de dados individual;
- Facilidade de Uso;
- Disponibilidade;

3.2 THINGSPEAK

Esta é uma plataforma “*open data*”, ou seja, é possível utilizar de alguns compartilhamentos de dados através da plataforma. A partir de dados coletados por meio sensores a plataforma é capaz de integrar os mesmos com diversos serviços, sistemas e tecnologias de terceiros. Ela funciona por meio de canais que armazenam ou enviam dados. Canais estes que possuem um total de 12 campos, onde, 8 são para armazenar qualquer tipo

de dado, 3 para armazenar localização e 1 para armazenar o *status*. Com esses canais o ThingSpeak pode suportar algumas funcionalidades de um sistema IoT, por exemplo, coletar dados recebidos por meio de sensores e enviá-los para alguma *cloud* onde eles possam ser melhor analisados e de acordo com o resultado acionar o funcionamento de uma aplicação. Algumas características dessa plataforma:

- Proporciona uma coleta de dados em tempo real;
- Análise de dados utilizando a ferramenta MATLAB;
- Comunicação entre dispositivos;
- Suporte na API;
- Fornece dados de geolocalização;

3.3 MICROSOFT AZURE IOT HUB

De acordo com a Microsoft, IoT atualmente é “*Internet of Your things*” e se inicia na própria empresa, por exemplo, adicionar dispositivos e serviços na infraestrutura local com intuito de uma expansão tecnológica para aumentar a visão em relação a empresa e ajudar em tomadas de decisões importantes. A plataforma oferece o serviço “*Microsoft Intelligent System Service*” que forma uma plataforma capaz de construir sistemas e aplicativos IoT por meio de coleta, armazenamento e processamento de dados.

Além disso, os serviços da Microsoft Azure possuem ferramentas como *Power BI* e *Office 365* que ajudam a organização a se conectar e gerenciar suas informações valiosas de maneira segura. Microsoft Azure IoT Hub ainda trabalha com *cloud* e possibilita uma solução com armazenamento, processamento e análise de dados. A plataforma tem a habilidade de transformar o negócio do cliente em uma solução flexível e inteligente. Algumas características dessa plataforma:

- Constrói sobre o que já foi estabelecido;
- Obtém mais benefícios de seus ativos existentes;
- Mudanças pequenas impacto grande;
- Suporte confiável;
- Escalabilidade;
- Conectividade;

3.4 GOOGLE CLOUD PLATAFORM

Nessa plataforma os desenvolvedores podem testar e fazer *deploy* de suas aplicações com alta escalabilidade e a certeza de estarem usando uma infraestrutura confiável, uma vez que ela é fornecida pelo Google e a própria empresa a utiliza. É uma das mais populares quando se trata de plataforma IoT por causa de sua rede extremamente rápida, sua ferramenta de *BigData* e o suporte a diversos tipos de serviços de nuvem como *RiptideIO*, *BigQuery*, conexão de Arduino, *Firebase*, *Cassandra* no Google Cloud Plataform, entre outros. Algumas características dessa plataforma:

- Funciona na infraestrutura da própria Google;
- Escalabilidade;
- Processamento, armazenamento e serviços;
- Performance;
- Ambiente de nuvem seguro;

3.5 IBM BLUEMIX

É uma PaaS desenvolvida pela IBM. Suporta as mais famosas linguagens de programação, como, *Java*, *PHP*, *Python*, *NodeJS*, entre outras. Com um conceito de DevOps integrado, permite realizar *build* e *deploy*, como também manusear as aplicações por meio da nuvem. A plataforma BlueMix é baseada na tecnologia *Cloud Foundry*, que é uma infraestrutura de camada suave. Por ser um produto da IBM é acompanhada por diversas outras tecnologias, como por exemplo, *IBM DataPower Gateway*, *IBM WebSphere Application Liberty Core*, entre outros. Algumas características dessa plataforma:

- Interface *Web* poderosa;
- Registro de dispositivo;
- Conectividade escalável;
- Segurança na comunicação;
- Suporte fornecido, caso necessário;

3.6 THINGWORX

É a primeira plataforma de *software* projetada para construir e operar o mundo conectado. Seu foco é reduzir o tempo, custo e risco presentes ao criar aplicações IoT.

Ela acelera o desenvolvimento de uma aplicação ao comprimir o ciclo *design-develop-deploy*. A plataforma permite o rápido desenvolvimento de aplicações inteligentes como, *Smart Agriculture, Smart Cities, Smart Grid, Smart Water, Smart Buildings e Telematics*. Algumas características dessa plataforma:

- Moderna e completa;
- Rápido *deploy*;
- Integração pessoa, sistema e máquina;
- Permite a aplicação evoluir e crescer com o tempo;
- Pesquisa inteligente;
- Permite colaboração;

3.7 DIGITAL SERVICE CLOUD

É uma plataforma recomendada para “*start-ups*”, pois permite que os desenvolvedores sejam donos de seus clientes da mesma forma que os clientes sejam donos de seus produtos. A plataforma tem o propósito de acelerar o processo de mercado para uma inovação. Utiliza a infraestrutura da própria Digital Service Cloud e por meio dela possibilita o monitoramento e gerenciamento do produto, além de conectá-lo a uma rede de milhões de dispositivos orientada pela interface. Algumas características dessa plataforma:

- Escalabilidade;
- Segurança;
- *Plug and Play*;
- Análise de *Big data*;
- Monitoramento, manutenção e gerenciamento de dispositivos;

3.8 ZETTA

É uma plataforma utilizada para criação de servidores IoT que funcionam em servidores geodistribuídos e em nuvem, ela combina REST API, programação reativa e “*web sockets*” o que é uma combinação muito forte quando se trata de montar muitos dispositivos em aplicativos intensivos de dados em tempo real.

O Zetta consegue rodar em uma ampla variedade de ambientes, seja computadores pessoais, computadores de placa única, nuvem, *Raspberry Pis, BeagleBones*, entre outros. Ao se comunicar com microcontroladores como Arduino, o Zetta pode fornecer a cada

dispositivo uma API REST localmente e também na nuvem. Fornece suporte para montagem de um sistema distribuído de dispositivos que se comunicam por meio da API e acesso direto aos protocolos e convenções subjacentes para que os desenvolvedores possam transformar sensores, atuadores e controladores de maneira fácil e eficiente em sistemas IoT inovadores.

A plataforma também permite monitorar o comportamento do dispositivo e do sistema e usar ferramentas para obter *insights* acionáveis e oferece *streaming* de dados em plataformas de análise de máquina, como a *Splunk*.

Em resumo o Zetta permite reunir aplicativos para *smartphones*, dispositivos e nuvem em sistemas grandes e complexos. Alguns exemplos são: automação residencial, transporte inteligente e computação vestível. Algumas características dessa plataforma:

- Compatibilidade;
- Proporciona uma API para “coisa”;
- Ambiente amigável de desenvolvimento;
- Fluxo de dados junto com conhecimento de dados;
- Suporte de desenvolvimento de grandes aplicativos;

3.9 NIMBITS

PaaS que pode ser instalada em qualquer dispositivo como, *Raspberry Pi*, *Web Server*, *Amazon EC2*, entre outros. É capaz de desenvolver aplicativos que podem se conectar uns com os outros ou com a nuvem e distribuir um conjunto grande de dados capaz de iniciar um evento ou disponibilizar uma análise de complexidade. Resolve a complexidade associada a *edge computing*, facilitando uma plataforma que é construída sobre o sistema local, executando um mecanismo de regras e enviando dados que são importantes para a nuvem. Algumas características dessa plataforma:

- *Download* dos servidores em *chips* ou na nuvem;
- Plataforma *open-source*;
- Facilidade em criar gatilhos que disparam eventos;
- Processar dados geodistribuídos e com *log*;

3.10 AMAZON WEB SERVICES

O Amazon Web Services (AWS) permite o conceito de IoT em escala global, facilitando os serviços prestados, a segurança e o suporte. Por ser uma rede grande, ele

é capaz de integrar alguns serviços da própria Amazon, como, *Amazon Elastic Cloud Compute (EC2)*, *Amazon Kinesis*, *Amazon Cognito*, *Amazon Mobile Analytics*, entre outros. A plataforma facilita a análise de *big data* e oferece suporte a dados de alto volume. Com o *Amazon Kinesis* é possível capturar com segurança dados de milhares de sensores. A AWS oferece diversos modelos pré-pagos para aplicativos IoT. A plataforma também oferece flexibilidade para os aplicativos IoT em termos de ferramentas, linguagem de programação, gerenciamento de dados e outros recursos de infraestrutura. Identidade do usuário, análise de dispositivos e notificações dos dispositivos são partes comuns de um aplicativo de Internet das Coisas. Algumas características dessa plataforma:

- Escalabilidade;
- Privacidade e Segurança;
- Modelo Pré-pago;
- Disponibilidade e Flexibilidade;
- Análise de dados e armazenamento de grande volume de dados;
- Fornece recursos de *hardware*;

A tabela de comparação Tabela 1 mostra a comparação de vários parâmetros entre as plataformas discutidas. As abreviaturas utilizadas nesta tabela são: A-Alto, M-Médio e B-Baixo. Foram utilizados vários parâmetros para comparação como facilidade de uso, escalabilidade, disponibilidade, tipo de solução, segurança, suporte para dados em tempo real e armazenamento de dados.

	Escalabilidade	Disponibilidade 24/7	Dados em Tempo Real	Armazenamento de Dados	Segurança e Privacidade	Facilidade de Uso	Tipo de Solução
Tago.IO	✓	✓	✓	✓	✓	A	PaaS
ThingSpeak	✓	✓	✓	X	✓	M	<i>IoT Analytics</i>
Azure IoT Hub	✓	✓	✓	✓	✓	B	PaaS
Google Cloud	✓	✓	✓	✓	✓	A	PaaS
IBM BlueMix	✓	✓	✓	✓	✓	A	PaaS
ThingWorx	✓	✓	✓	✓	✓	M	<i>Complete IoT</i>
Digital Service Cloud	✓	✓	✓	X	✓	M	PaaS
Zetta	X	✓	✓	X	X	A	<i>API to Devices</i>
Nimbits	✓	✓	✓	✓	X	B	PaaS
Amazon Web Service	✓	✓	✓	✓	✓	M	IaaS

Tabela 1: Comparação entre as plataformas

A Tabela 1 foi produzida através da leitura de alguns artigos, como, Nakhuva et al (2015), IoTDesignPro (2019), Ray (2016), Babun et al (2019), Bertoleti (2020) e pesquisas nos *sites* das plataformas, como, Tago.IO (2021), ThingSpeak (2021), Microsoft (2021), entre outros. A escolha das plataformas para a realização do estudo de caso foi feita com base no tamanho da marca, na facilidade de uso e na diferença de implementação entre cada uma.

4 ESTUDO DE CASO

4.1 IMPLEMENTAÇÃO BÁSICA EM PLATAFORMAS IOT

Como estudo de caso teve-se a ideia de monitorar um sistema de temperatura onde, através de um sensor, seria possível em um determinado intervalo de tempo obter a temperatura e umidade de um local e enviar esses dados para as plataformas.

Com os dados no sistema é possível realizar um teste de como é o gerenciamento das informações enviadas, se existe algum *dashboard*, qual o tempo de atualização dessa interface, como é feito a execução de códigos pela plataforma e estudar o poder da plataforma para realização de análises em cima dos dados e a execução de algumas funções sobre determinadas condições.

Para simular o sensor será utilizado o auxílio de uma aplicação externa, onde através de uma chamada HTTP (*Hypertext Transfer Protocol*) simples será possível obter condições climáticas de uma região em tempo real. Com esses dados prontos é necessário enviá-los para a plataforma e isso pode ser feito utilizando o protocolo MQTT.

4.2 PLATAFORMA TAGO.IO



Figura 5: Tago.IO

Uma das plataformas que será utilizada para um caso de teste é a Tago.IO¹. Ela é muito interessante para aqueles que precisam desenvolver um sistema de demonstração, pois possui planos gratuitos e pagos. Os planos gratuitos são bem robustos e permitem a realização de protótipos avançados. A comunicação é feita por meio dos protocolos mais famosos, como, MQTT e HTTPS (*Hyper Text Transfer Protocol Secure*). É ideal para criar *dashboards*, disparar ações automáticas, rodar análises periódicas automaticamente via *scripts* desenvolvidos em *NodeJS* e permite armazenar os dados em *buckets* específicos,

¹ <https://tago.io/>

possuindo diversos tipos de disparo de ações, como, envio de *e-mail*, mensagem SMS (*Short Message Service*), mensagem MQTT, entre outros.

Para utilização da plataforma é necessário primeiramente realizar um cadastro em Tago.IO e configurar um novo dispositivo, que será responsável por capturar e enviar os dados para a mesma.

Após o cadastro, é possível através da própria interface criar os dispositivos e configurar qual será o protocolo de comunicação com ele, como mostra a Figura 6. Na Tago.IO cada dispositivo conta com uma *token* única responsável pela autenticação, conforme ilustrado na Figura 7.

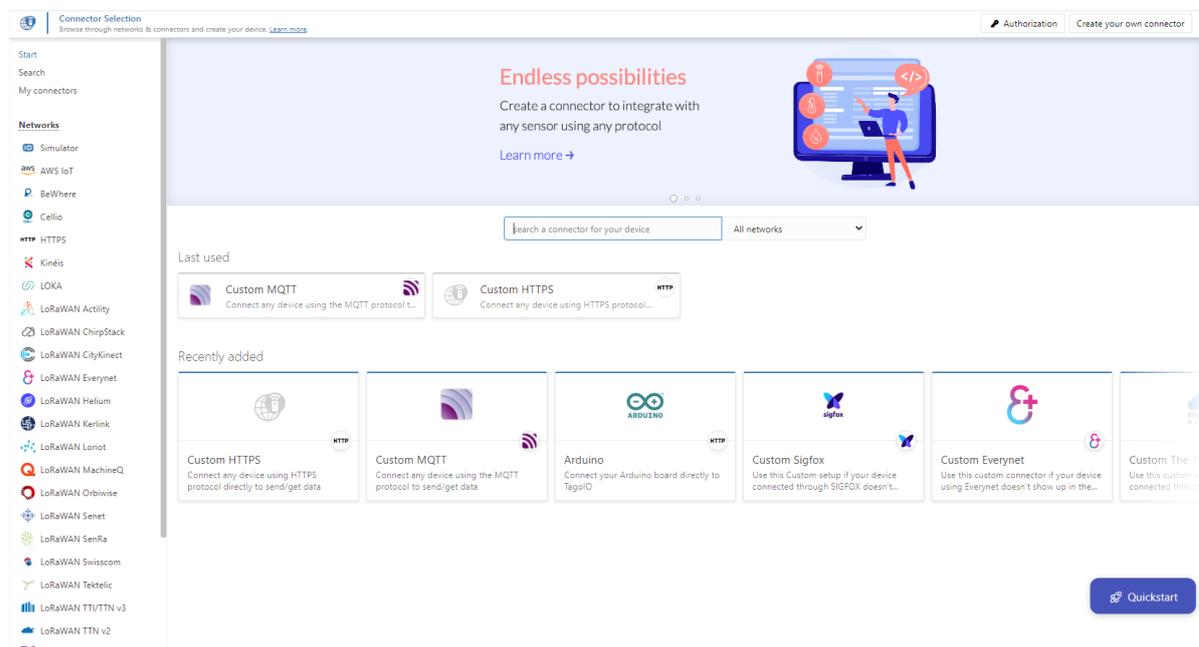


Figura 6: Criação de dispositivos na Tago.IO

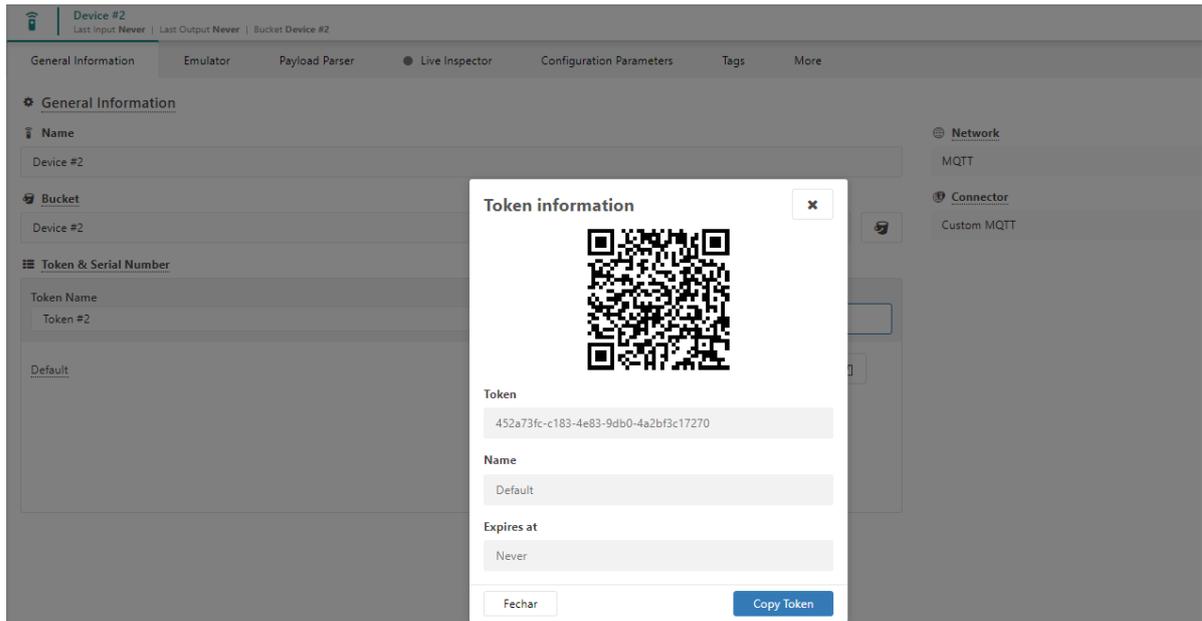


Figura 7: Dispositivo na Tago.IO

A comunicação será feita com o protocolo MQTT, sendo assim, primeiramente será estabelecida uma conexão com o *broker*(*mqtt.tago.io*) e porta 8883 da Tago.IO e o dispositivo será autenticado por meio de sua *device-token* única. Após isso é necessário se inscrever em um tópico para enviar as mensagens com as informações das variáveis coletadas pelo sensor, conforme ilustrado nas Figuras 8, 9 e 10.

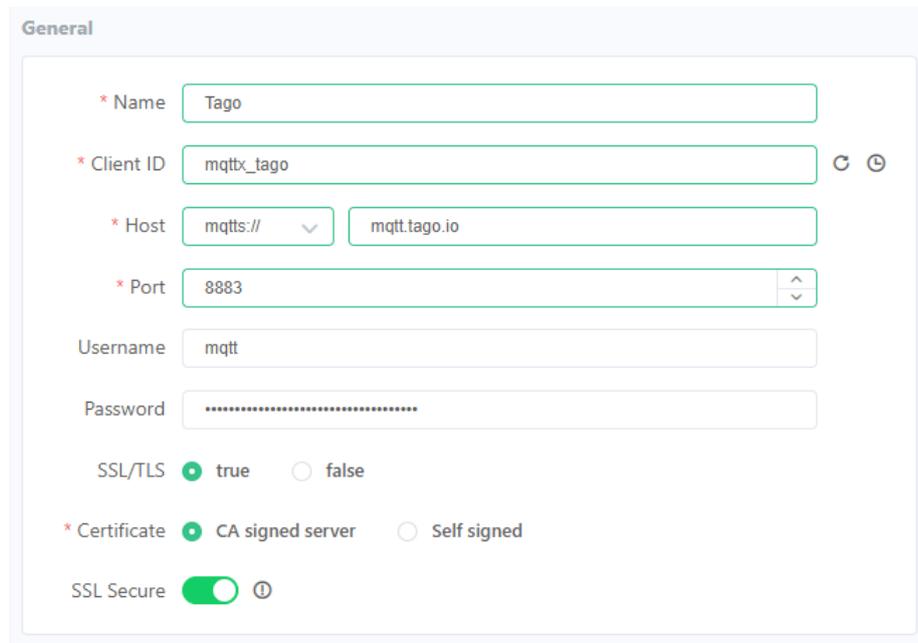


Figura 8: Conectando ao *broker*

New Subscription

* Topic
tago/my_topic

* QoS Color
0 #A8F19A

Alias

Cancel Confirm

Figura 9: Inscrevendo no tópic

Payload: JSON QoS: 0 Retain

tago/my_topic

```
[
  {
    "variable": "temperature",
    "value": 30
  }
]
```

Figura 10: Enviando informação para o tópic

Com os dados já na plataforma podemos utilizar seus recursos para aplicar eventos em cima de determinadas variáveis, por exemplo, obter o valor máximo, valor mínimo e a média de uma variável. Esses valores podem ser obtidos através de um *script* feito em *NodeJS* aplicado em cima de um *bucket* de um dispositivo. Além disso, possibilita enviar automaticamente *e-mails* e *SMSs* caso a informação atinja determinadas condições que devem ser configuradas pela própria UI, como mostrado na Figura 11.

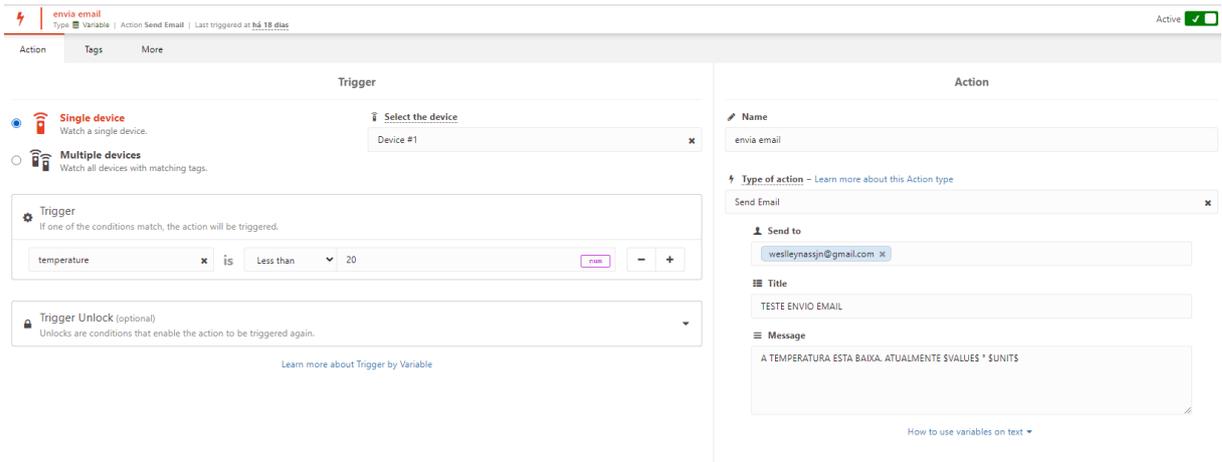


Figura 11: *Action* para envio e-mail

A Tago.IO proporciona a criação de múltiplos *dashboards* com painéis totalmente configuráveis e é capaz de exibir as informações em tempo real, mantendo o usuário com maior controle dos seus dados, como ilustrado na Figura 12.



Figura 12: *Dashboard* Tago.IO

Um ponto muito positivo da Tago.IO é que ela disponibiliza tudo através de uma interface gráfica de usuário, o que deixa o requisito não funcional de facilidade de uso bem elevado. Além disso, ela disponibiliza planos gratuitos com bastante diversidade, como visto na Figura 13. Portanto, para projetos pequenos a Tago.IO é uma ótima escolha.

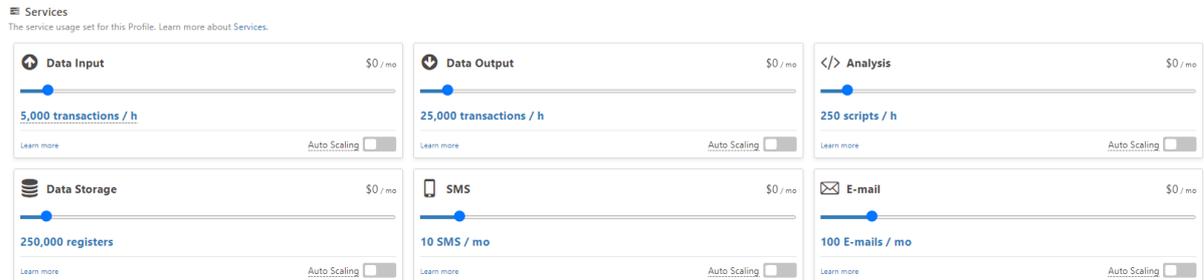


Figura 13: *Services* Tago.IO

4.3 PLATAFORMA THINGSPEAK



Figura 14: ThingSpeak

A ThingSpeak² é uma plataforma *open-source* e permite que uma instância sua seja criada em qualquer servidor que deseja utilizá-la para fazer uso de uma forma mais fechada ou prioritária. Também suporta os protocolos de comunicação HTTP e MQTT, os mais populares do mercado de IoT.

Ideal para monitorar dados históricos e de longo prazo. Seu maior diferencial é a integração nativa com o MATLAB, uma ferramenta poderosa para operações matemáticas, ou seja, é possível criar e executar *scripts* responsáveis por resolver modelos matemáticos complexos por trás da plataforma. Possui também integração com o Twitter, sendo assim, pode-se gerar *tweets* automáticos sob efeito de gatilhos.

Por ser uma plataforma que utiliza canais para fazer a comunicação entre dispositivo e usuário, primeiramente é necessário criar um canal e associar um dispositivo a ele, isso pode ser feito pela própria UI, como ilustrado nas Figuras 15 e 16.

² <https://thingspeak.com/>

New Channel

Name

Description

Field 1

Field 2

Field 3

Field 4

Field 5

Field 6

Field 7

Field 8

Figura 15: *New Channel* ThingSpeak

Add a new device ×

Device Information

Name*

Description

Authorize channels to access ⓘ

▾

Authorized Channel ⓘ	Allow Publish	Allow Subscribe
No channels authorized.		

Figura 16: *Create Device* ThingSpeak

Para envio dos dados, é necessário a conexão com o seu *broker*, para isso as seguintes informações serão utilizadas, como ilustrado na Figura 17:

- *Client Id*: {CLIENT_ID_DIPOSITIVO}
- *Host*: mqtt://mqtt3.thingspeak.com
- *Port*: 1883
- *Username*: {USERNAME_DISPOSITIVO}
- *Password*: {PASSWORD_DISPOSITIVO}

The image shows a 'General' configuration window for a ThingSpeak connection. It contains the following fields and options:

- Name:** ThingSpeak
- Client ID:** CysUNJQkHTckEDEPKggPEAs
- Host:** mqtt:// (dropdown), mqtt3.thingspeak.com
- Port:** 1883
- Username:** CysUNJQkHTckEDEPKggPEAs
- Password:** [masked]
- SSL/TLS:** true, false

Figura 17: *Connection* ThingSpeak

Após a conexão com o *broker* será necessário se inscrever no tópico que será o canal de comunicação e troca de “mensagens” do dispositivo para a plataforma, como ilustrado na Figura 18. É necessário informar o identificador do canal criado anteriormente no *site*.

The image shows a 'New Subscription' dialog box with the following configuration:

- Topic:** channels/1478314/subscribe
- QoS:** 0
- Color:** #A5FB8B
- Alias:** [empty field]
- Buttons:** Cancel, Confirm

Figura 18: *Subscribe Topic* ThingSpeak

Com a inscrição feita, por fim a mensagem MQTT será enviada através de *Plaintext* por meio do tópico de publicação de dados. Essa mensagem deve conter as variáveis com seus respectivos valores e um *status* informando que se trata de um MQTTPUBLISH, como ilustrado na Figura 19.

The image shows the MQTT client interface with the following configuration:

- Payload:** Plaintext
- QoS:** 0
- Retain:**
- Topic:** channels/1478314/publish
- Payload:** field1=15&status=MQTTPUBLISH

Figura 19: *Topic & payload* ThingSpeak

Logo na página inicial do canal temos um *dashboard* disponível que permite a visualização das mensagens recebidas e outras informações, como, média, gráficos de

variação, entre outros, como visto na Figura 20. Estas variáveis são atualizadas no intervalo de tempo de 15 segundos devido a limitação do plano gratuito da plataforma.

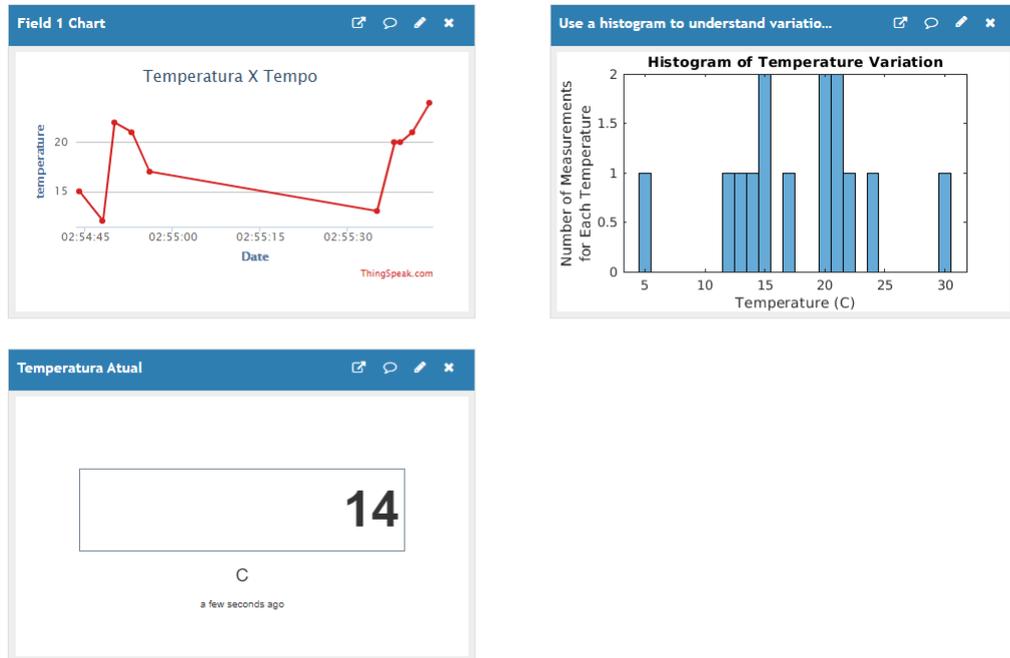


Figura 20: *Dashboard* ThingSpeak

A média pode ser calculada por meio de uma *MATLAB Analysis*, onde o desenvolvedor escreve um código para obter as temperaturas em um intervalo de tempo e realizar o cálculo do seu valor máximo, mínimo e médio. Feito isso, dentro do código é possível atualizar o *dashboard* do canal e visualizar essas informações. Essa análise pode ser acionada com uma condição de variável ou de tempos em tempos.

O envio de *e-mail* também é feito por meio de uma *MATLAB Analysis* e é ocorre através de um *endpoint* da própria ThingSpeak, portanto, o *e-mail* é enviado para a conta que está logada no momento. Uma *MATLAB Analyses* pode ser acionada por meio de uma *React*, que é engatilhada quando um canal disponível recebe alguma informação e esta atinge certa condição, conforme Figura 21.

Apps / React / React 1

[Edit React](#)

Name:	React 1
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2021-08-18 23:02
Channel:	temp
Condition:	Field 1 (temperature) is less than 20
MATLAB Analysis:	Read Channel to Trigger Email 1
Run:	Each time the condition is met
Created:	2021-08-18 10:56 pm

Figura 21: *React* ThingSpeak

Possui um plano gratuito e um pago, sendo assim, algumas limitações devem ser respeitadas caso a escolha seja o gratuito, conforme Figura 22.

	FREE For time-limited commercial evaluation of the service	STANDARD For all commercial, government and revenue generating activities
Scalable for larger projects	✘ No. Annual usage is capped.	✔
Number of messages	3 million/year (~8,200/day) ⁽¹⁾	33 million/year per unit (~90,000/day per unit) ⁽²⁾
Message update interval limit	Every 15 seconds	Every second
Number of channels	4	250 per unit
MATLAB Compute Timeout	20 seconds	60 seconds
Private channel sharing	Limited to 3 shares	Unlimited
Technical Support	Community Support	Standard MathWorks support

⁽¹⁾Based on constant update rate.

Pricing calculator

How many channels?*

Currently:

To be added:

Calculated number of channels needed: 10

Purchase

License type: **Standard**

ThingSpeak units:

x USD 650.00 price/unit/year

Total: USD 650.00/year

[Purchase](#)

Figura 22: *Services* ThingSpeak

4.4 PLATAFORMA AZURE IOT HUB

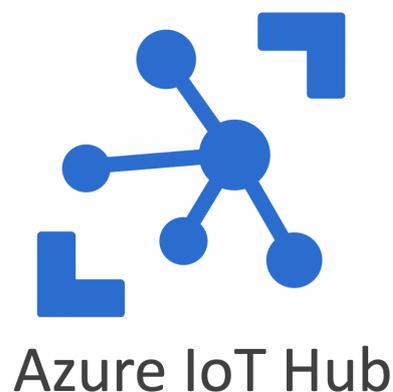


Figura 23: Azure IoT Hub

Mais uma plataforma de grande influência no mercado é a Azure IoT Hub³. Ela é muito poderosa quando o desenvolvimento se trata de um sistema no qual será necessário a manipulação de um volume muito alto de dados. Além dos serviços oferecidos, como o de se conectar e monitorar bilhões de ativos geograficamente distribuídos, a plataforma inclui sistemas operacionais e segurança de ponta para os dispositivos e equipamentos, bem como dados e análises que ajudam outras empresas a criar, implantar e gerenciar seus aplicativos. A marca dá ênfase dizendo que quando se trata de computação em nuvem e computação de borda, os poderes computacionais e oportunidades de sistemas autônomos são cada vez maiores.

Para desenvolver o estudo de caso, primeiramente deve-se cadastrar no portal da Azure, diferente das outras plataformas testadas esse cadastro deve ser realizado utilizando um número de celular e um cartão de crédito independente do plano escolhido. Após o cadastro, será necessário primeiro criar um “*Resource*”, uma instância do “*Iot Hub*”, escolhendo um nome para o mesmo, como visto na Figura 24, logo em seguida, na aba “*Iot devices*” um dispositivo poderá ser configurado, conforme Figuras 25 e 26.

³ <https://azure.microsoft.com/pt-br/services/iot-hub/>

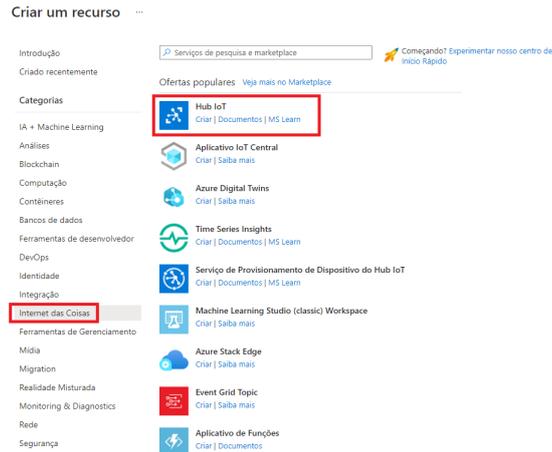


Figura 24: Criação *Resource* Azure

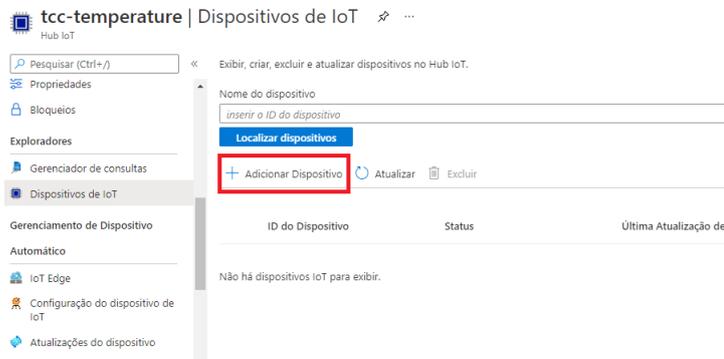


Figura 25: Criação dispositivo Azure

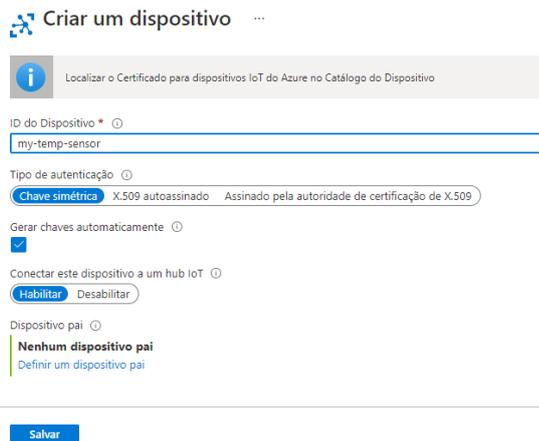


Figura 26: Configurar novo dispositivo Azure

Além do dispositivo é necessário obter uma *password* responsável por autenticar o dispositivo que realizará a comunicação via MQTT. Na opção Políticas de acesso compartilhado e em *iothubowner* pode-se obter a *Connection String* que será utilizada pelo *software* auxiliar *Device Explorer Twin* para gerar esta *password* (`{TOKEN_DEVICE}`), como visto nas Figuras 27 e 28.



Figura 27: *Connection String* Azure

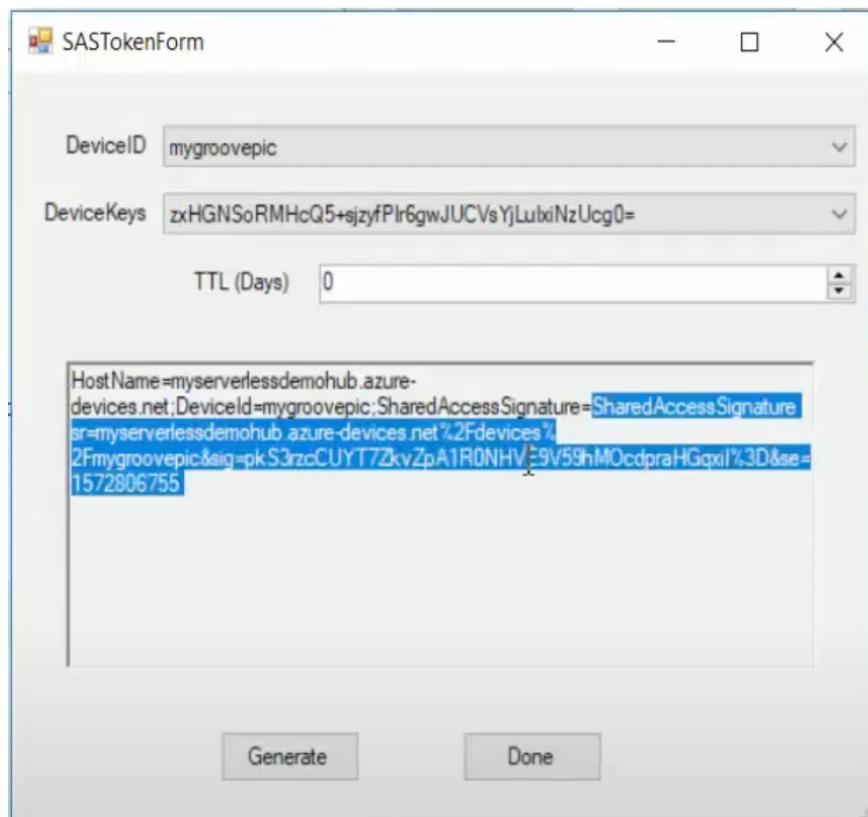


Figura 28: Geração *token* Azure

Feito isso, é necessário preparar os dados para uma conexão com o *broker* MQTT, conforme ilustrado na Figura 29. Basicamente precisamos dos seguintes itens:

- *Url*: `ssl://{NOME_RESOURCE}.azure-devices.net`
- *Port*: 8883

- *ClientId*: {NOME_DISPOSITIVO}
- *Username*: {NOME_RECURSO}.azure-devices.net/{NOME_DISPOSITIVO}
- *Password*: {TOKEN_DEVICE}
- *Topic*: devices/{NOME_DISPOSITIVO}/messages/events/{EXTENSAO_TOPICO}

The screenshot shows a 'General' configuration window with the following fields and values:

- Name**: Azure IoT Hub
- Client ID**: my-temp-sensor
- Host**: mqts:// tcc-temperature.azure-devices.net
- Port**: 8883
- Username**: tcc-temperature.azure-devices.net/my-temp-sensor
- Password**: [Masked]
- SSL/TLS**: true false
- Certificate**: CA signed server Self signed
- SSL Secure**:

Figura 29: *Connection broker* Azure

Será preciso realizar a inscrição no tópico e criar um JSON (*JavaScript Object Notation*) para enviar os dados, conforme ilustrado nas Figuras 30 e 31.

The screenshot shows a 'New Subscription' dialog box with the following fields and values:

- Topic**: devices/my-temp-sensor/messages/events/temperature
- QoS**: 0
- Color**: #27F24E
- Alias**: [Empty]

Buttons: Cancel, Confirm

Figura 30: Inscrição tópico Azure

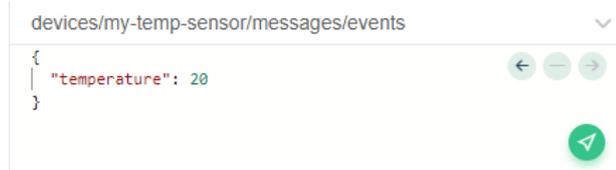


Figura 31: Enviando informação para o tópic

Com todas as variáveis prontas para executar a comunicação, pode-se elaborar uma aplicação simples utilizando a mesma ideia da plataforma anterior, sendo assim, também será simulado um sensor de temperatura, e a plataforma será utilizada para gerar as médias, máximo e mínimo da variável e enviar um *e-mail* sobre determinada condição.

Para executar o envio de *e-mail* é necessário que seja configurada uma *Azure Function*. Essa função pode ser implementada pela própria IDE da Microsoft, o Visual Studio⁴, ela será responsável por configurar um servidor SMTP (*Simple Mail Transfer Protocol*), responsável por fazer o envio do *e-mail* quando a condição for atingida, como ilustrado na Figura 34. É necessário também obter uma *key* para configurar essa função criada com o Azure IoT Hub, ela pode ser obtida na aba Pontos de extremidade externas, como visto na Figura 32 e seu valor deve ser adicionado ao arquivo de configuração da *Azure Function*, como ilustrado na Figura 33.

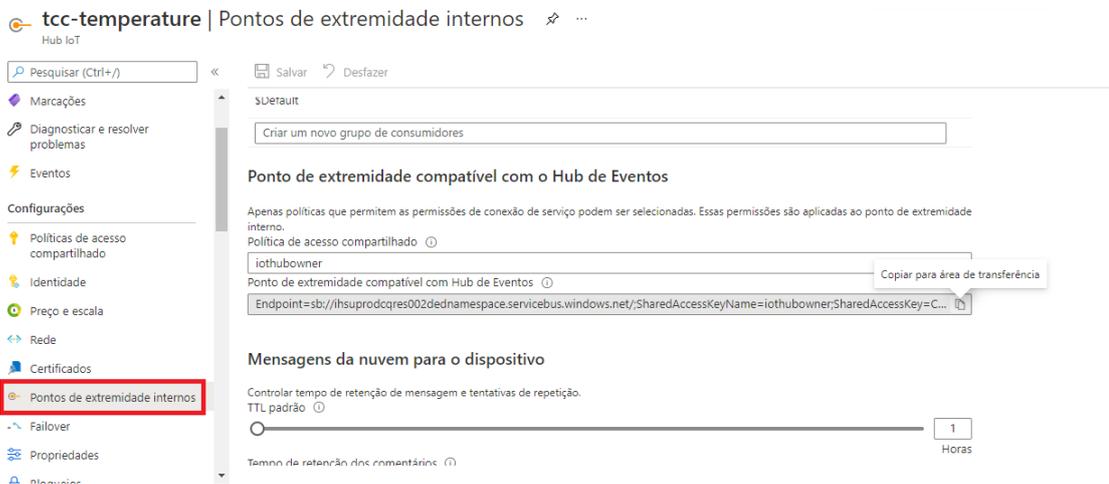


Figura 32: *Endpoint Key Azure Function*

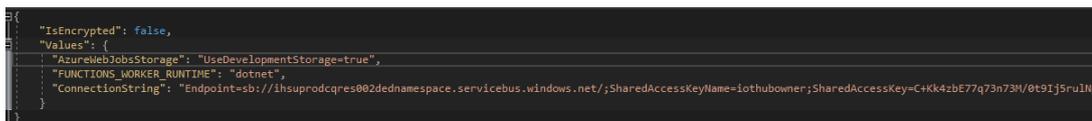


Figura 33: *Local Settings Function*

⁴ <https://visualstudio.microsoft.com/>

```

public static void SendEmail(string temperature)
{
    MailMessage mailMessage = new MailMessage("EMAIL_ORIGEM", "EMAIL_DESTINO");
    mailMessage.Body =
        $"Temperatura atual de {temperature}°C está abaixo da condição estabelecida e disparou o envio de email.";
    mailMessage.Subject = "Temperatura Baixa Azure IoT Hub";

    SmtpClient smtpClient = new SmtpClient(host: "smtp.gmail.com", port: 587);
    smtpClient.Credentials = new System.Net.NetworkCredential()
    {
        UserName = "EMAIL_ORIGEM@gmail.com",
        Password = "SUA_PASSWORD"
    };
    smtpClient.EnableSsl = true;

    try
    {
        smtpClient.Send(mailMessage);
    }
    catch (Exception e)
    {
        Console.WriteLine(e);
        throw;
    }
}

```

Figura 34: *Azure Function*

Como as outras plataformas, a Azure IoT Hub também disponibiliza um *dashboard* para um acesso mais fácil as informações dos dispositivos, como apresentado na Figura 35.

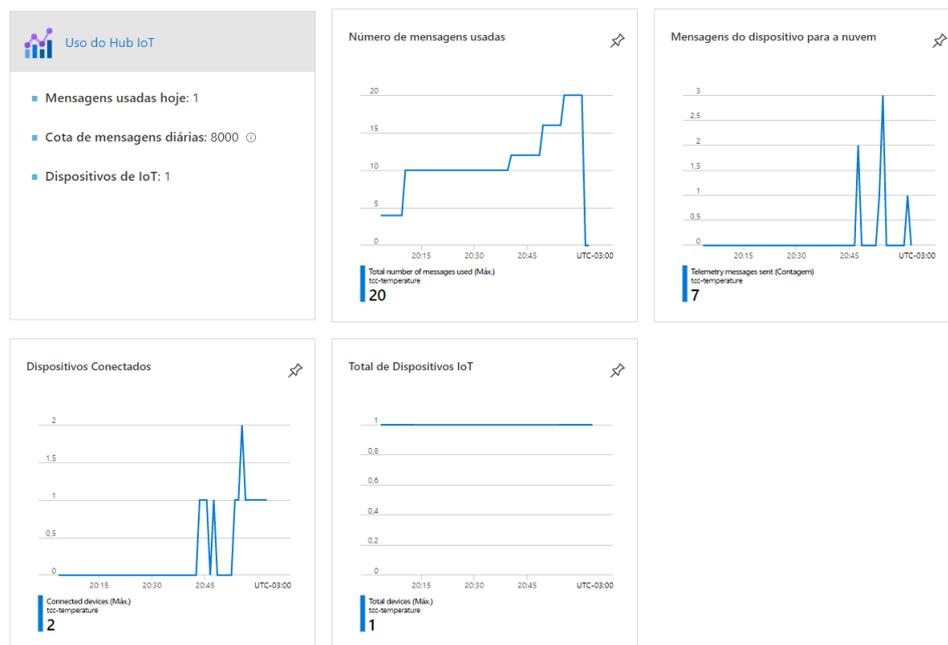


Figura 35: *Dashboard Azure*

Diferente das outras plataformas, essa obriga a inserção de um cartão de crédito ao realizar o cadastro, o que acaba atrapalhando um pouco quando os projetos tendem a ser desenvolvidos para aquisição de conhecimento ou apenas para serem usados como protótipos. Possui dois planos e cada um com suas limitações, onde quanto mais recursos forem utilizados, maior será seu valor, como visto na Figura 36.

Camada básica

Tipo de edição	Preço por unidade de Hub IoT (por mês)	Número total de mensagens/dia por unidade de Hub IoT	Tamanho do medidor de mensagem
B1	R\$48,989	400.000	4 KB
B2	R\$244,945	6.000.000	4 KB
B3	R\$2.449,450	300.000.000	4 KB

Camada Standard

Tipo de edição	Preço por unidade de Hub IoT (por mês)	Número total de mensagens/dia por unidade de Hub IoT	Tamanho do medidor de mensagem
Gratuito	Gratuito	8.000	0,5 KB
S1	R\$122,473	400.000	4 KB
S2	R\$1.224,725	6.000.000	4 KB
S3	R\$12.247,25	300.000.000	4 KB

Figura 36: *Services Azure*

5 CONCLUSÃO E TRABALHOS FUTUROS

A necessidade de escolher uma plataforma antes de desenvolver o sistema desejado vem sendo cada vez mais necessário devido ao crescimento da IoT e o aumento de seu uso em diversas áreas da computação. Hoje em dia, a variedade de soluções disponíveis não param de aumentar e seus objetivos são praticamente os mesmos, porém o propósito de cada uma, apesar de ser muito parecido, se difere pelo fato de que cada uma se destaca em pontos diferentes. A finalidade desse trabalho foi mostrar que dentre as diversas plataformas do mercado cada uma tem um ponto que pode ser considerado como seu atrativo.

No caso da Tago.IO, foi notado que ela permite diversas operações de forma gratuita e é uma plataforma muito intuitiva onde é possível fazer tudo na própria interface, como, criar múltiplos dispositivos, criar novos *buckets*, executar *scripts* em cima desses dados, disparar gatilhos, monitorar dados via *dashboards*, entre outros. Porém, mesmo em seus planos pagos, a mesma não apresenta nenhum tipo de *backup* para os dados quando ocorre uma falha de conexão. É uma plataforma indicada para o desenvolvimento de uma monografia por conta de sua flexibilidade em seus planos gratuitos.

A ThingSpeak, assim como a Tago.IO, deixa que o usuário configure todas suas informações pela própria interface o que a torna uma plataforma fácil de usar, porém, diferente das outras exige um conhecimento breve do MATLAB para criar suas análises. Entretanto, tendo esse conhecimento, ela acaba sendo uma plataforma muito poderosa quando se é necessário desenvolver aplicações que exigem modelos matemáticos mais complexos. Assim como as outras, ela também disponibiliza *dashboards* e gatilhos para eventos. Sendo assim, é uma plataforma indicada para criar soluções que usem do MATLAB para gerar gráficos e análises matemáticas mais complexas.

A Azure IoT Hub é a mais complicada de se usar e obriga a inserção de um cartão de crédito tornando a exploração mais difícil. Seus planos gratuitos só permitem a criação de uma instância, obrigando que dispositivos distintos permaneçam na mesma instância. Todos os seus gatilhos devem ser feitos por meio de *Azure Functions*, exigindo implementação de todos, porém, ela também permite monitoramento via *dashboard*, múltiplos dispositivos na mesma instância, análises via código em cima dos dados, disparo de gatilhos e é a que mais se preocupa no quesito de segurança e tratamento de um volume grande de dados. Portanto, essa plataforma é indicada para soluções maiores que visam a confidencialidade, integridade e autenticidade dos seus dados.

Em trabalhos futuros, podem ser feitos testes de desempenho nas plataformas, como, medir tempo de requisições ou utilizar uma quantidade massiva de dados. Também é possível explorar mais plataformas do mercado e testar os conceitos de *Edge Computing*, onde os sensores são responsáveis por um pré-processamento dos dados, para desta forma

ver como o serviço vai se comportar.

REFERÊNCIAS

- Babun, L.; Denney, K.; Celik, Z. B.; McDaniel, P. ; Uluagac, A. S. A survey on iot platforms: Communication, security, and privacy perspectives. *Computer Networks*, v.192, p. 108040, 2021.
- Barbosa, K. G.; Teixeira, I. T. T. ; Behrens, F. Implementação de aplicações de monitoramento em internet das coisas. 2017.
- BERTOLETI, P. O que são as plataformas iot e quais são as maiores do mercado (mic435), 2020. Disponível em <https://www.newtoncbraga.com.br/index.php/iot/17606-o-que-sao-as-plataformas-iot-e-quais-sao-as-maiores-do-mercado-mic435.html>.
- MEIRELLES, F. S. 32ª pesquisa anual do uso de ti nas empresas, 2021. FGV EAESP, 2021.
- Fahmideh, M.; Zowghi, D. An exploration of iot platform development. *Information Systems*, v.87, p. 101409, 2020.
- Guth, J.; Breitenbücher, U.; Falkenthal, M.; Leymann, F. ; Reinfurt, L. Comparison of iot platform architectures: A field study based on a reference architecture. In: 2016 Cloudification of the Internet of Things (CIoT), p. 1–6. IEEE, 2016.
- HEUVELDOP, N. Ericsson mobility report. In: Ericsson Mobility Report, 2017.
- IoTDesignPro. Top 10 iot cloud platforms, 2019. Disponível em <https://iotdesignpro.com/articles/top-10-iot-cloud-platforms>.
- Lueth, K. Iot platform companies landscape 2019/2020: 620 iot platforms globally. *IoT Analytics*, Dec, 2019.
- MQTT.ORG. Mqtt: The standard for iot messaging, 2020. Disponível em <https://mqtt.org/>.
- MICROSOFT. Hub iot do azure, 2021. Disponível em <https://azure.microsoft.com/pt-br/services/iot-hub/#overview>.
- Nakhuva, B.; Champaneria, T. Study of various internet of things platforms. *International Journal of Computer Science & Engineering Survey*, v.6, n.6, p. 61–74, 2015.
- Pankaj, G. Selecting the right iot cloud platform. In: 2016 International Conference on Internet of Things and Applications (IOTA), p. 316–320. IEEE, 2016.
- Pires, P. F.; Delicato, F.; Batista, T.; Barros, T.; Cavalcante, E. ; Pitanga, M. Plataformas para a internet das coisas. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 110–169, 2015.
- Ray, P. P. A survey of iot cloud platforms. *Future Computing and Informatics Journal*, v.1, n.1-2, p. 35–46, 2016.
- Salami, A.; Yari, A. A framework for comparing quantitative and qualitative criteria of iot platforms. In: 2018 4th International Conference on Web Research (ICWR), p. 34–39. IEEE, 2018.

SCHULTZ, F. O que é internet das coisas (iot) e como funciona?, 2020. Disponível em <https://milvus.com.br/internet-das-coisas-iot/>.

das Dores Sousa, E.; de Oliveira, R. F. Emulador de publisher mqtt para redes de sensores sem fio. 2019.

TAGO.IO. Tago.io easily create your own iot solutions, 2021. Disponível em <https://tago.io/>.

THINGSPEAK. Thingspeak for iot projects, 2021. Disponível em <https://thingspeak.com>.

Torres, A. B.; Rocha, A. R. ; de Souza, J. N. Análise de desempenho de brokers mqtt em sistema de baixo custo. In: Anais do XV Workshop em Desempenho de Sistemas Computacionais e de Comunicação, p. 47–58. SBC, 2016.

Vandikas, K.; Tsiatsis, V. Performance evaluation of an iot platform. In: 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, p. 141–146, 2014.

YUAN, M. Conhecendo o mqtt, 2017. Disponível em <https://developer.ibm.com/br/articles/iot-mqtt-why-good-for-iot/>.